**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Выполнил: |  |  |  |  |
| студент группы ИУ5-33Б |  |  |  |  | |
|  |  |  |  |  |
| Алёшин Александр |  |  |  |  |

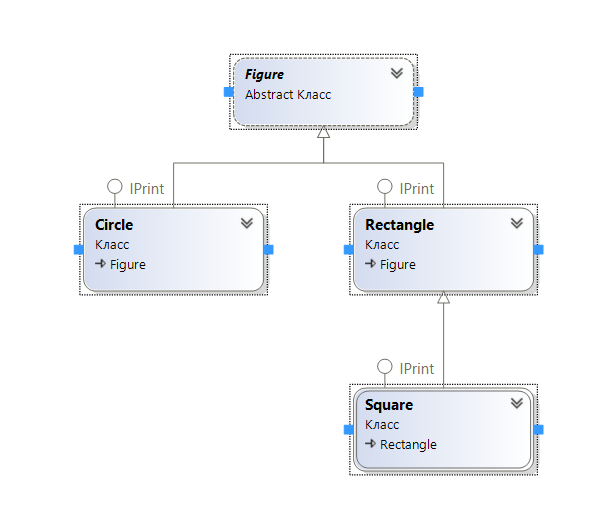
Москва, 2020 г.

Условие лабораторной работы:

Разработать программу, реализующую работу с классами.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Абстрактный класс «Геометрическая фигура» содержит виртуальный метод для вычисления площади фигуры.
3. Класс «Прямоугольник» наследуется от «Геометрическая фигура». Ширина и высота объявляются как свойства (property). Класс должен содержать конструктор по параметрам «ширина» и «высота».
4. Класс «Квадрат» наследуется от «Прямоугольник». Класс должен содержать конструктор по длине стороны.
5. Класс «Круг» наследуется от «Геометрическая фигура». Радиус объявляется как свойство (property). Класс должен содержать конструктор по параметру «радиус».
6. Для классов «Прямоугольник», «Квадрат», «Круг» переопределить виртуальный метод Object.ToString(), который возвращает в виде строки основные параметры фигуры и ее площадь.
7. Разработать интерфейс IPrint. Интерфейс содержит метод Print(), который не принимает параметров и возвращает void. Для классов «Прямоугольник», «Квадрат», «Круг» реализовать наследование от интерфейса IPrint. Переопределяемый метод Print() выводит на консоль информацию, возвращаемую переопределенным методом ToString().

Диаграмма классов:



Текст программы:

Figure.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

abstract class Figure

{

/// <summary>

/// Тип фигуры

/// </summary>

public string Type

{

get

{

return this.\_Type;

}

protected set

{

this.\_Type = value;

}

}

string \_Type;

/// <summary>

/// Вычисление площади

/// </summary>

public abstract double Area();

/// <summary>

/// Приведение к строке, переопределение метода Object

/// </summary>

public override string ToString()

{

return this.Type + " площадью " + this.Area().ToString();

}

}

}

Rectangle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

class Rectangle : Figure, IPrint

{

/// <summary>

/// Высота

/// </summary>

public double height { get; set; }

/// <summary>

/// Ширина

/// </summary>

public double width { get; set; }

/// <summary>

/// Основной конструктор

/// </summary>

/// <param name="ph">Высота</param>

/// <param name="pw">Ширина</param>

public Rectangle(double ph, double pw)

{

height = ph;

width = pw;

this.Type = "Прямоугольник";

}

/// <summary>

/// Вычисление площади

/// </summary>

public override double Area()

{

double Result = width \* height;

return Result;

}

public override string ToString()

{

return base.ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

}

Square.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

class Square : Rectangle, IPrint

{

public Square(double size) : base(size, size)

{

this.Type = "Квадрат";

}

public override string ToString()

{

return base.ToString();

}

public new void Print()

{

Console.WriteLine(this.ToString());

}

}

}

Circle.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

class Circle : Figure, IPrint

{

/// <summary>

/// Ширина

/// </summary>

public double radius { get; set; }

/// <summary>

/// Основной конструктор

/// </summary>

/// <param name="ph">Высота</param>

/// <param name="pw">Ширина</param>

public Circle(double pr)

{

radius = pr;

this.Type = "Круг";

}

public override double Area()

{

double Result = Math.PI \* radius \* radius;

return Result;

}

public override string ToString()

{

return base.ToString();

}

public void Print()

{

Console.WriteLine(this.ToString());

}

}

}

IPrint.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab2

{

interface IPrint

{

void Print();

}

}

Пример выполнения программы:

